# Meetrapport titel

## Namen en datum

Dit meetrapport is gemaakt op 14 april door Thijs Hendrickx en Job Verhaar

## Doel

Het doel van het experiment is om te kijken hoe een zo snel mogelijke imageshell gemaakt kan worden. Hierbij wordt gekeken naar de snelheid van het initialiseren van een image maar vooral naar snelheid van het opvragen van een pixelwaarde aangezien dat meer nodig is.

## Hypotheses

Door gebruik te maken van een 2d array kan het terugvragen van een pixel waarde versneld worden. Het initialiseren van een afbeelding zal slomer worden maar de snelheid verbetering van het terugkrijgen van een pixelwaarde weegt hier ruim tegen op.

## Werkwijze

Allereerst is er een extra klasse gemaakt voor het timen van een functie. Functies zijn als volgt:

class Clock {

public:

static timepoint now() {

return high\_resolution\_clock::now();

}

static double getMicroSecondsFrom(timepoint& p) {

return double(duration\_cast<microseconds>(now() - p).count());

}

static double getMilliSecondsFrom(timepoint& p) {

return double(duration\_cast<milliseconds>(now() - p).count());

}

static double getNanoSecondsFrom(timepoint& p) {

return double(duration\_cast<nanoseconds>(now() - p).count());

}

};

Om de snelheid van een functie getest moet worden kan vervolgens volgende worden gedaan:

auto timepointBefore = Clock::now();

//Execute function(s)

auto durationOfFunction = Clock::getNanoSecondsFrom(timepointBefore);

Het experiment dat uitgevoerd gaat worden is het testen van de snelheid van het initialiseren van een RGBImage. En het terugkrijgen van een bepaalde pixel. Beide functies worden een bepaald aantal keer uitgevoerd in een for loop en de resultaten zullen worden bekeken;

## Resultaten

Allereerst basis constructor getest met volgende functie RGBImage(int width, int height);

*Testing constructor timing of image with constructing image of 500x500 1000 times*

*Private image:*

*-Duration total: 338267944.0000000000 nanoseconds*

*-Duration average per image: 338267.9440000000 nanoseconds*

*Student image:*

*-Duration total: 440710291.0000000000 nanoseconds*

*-Duration average per image: 440710.2910000000 nanoseconds*

*Testing constructor timing of image with constructing image of 1000x1000 200 times*

*Private image:*

*-Duration total: 248749451.0000000000 nanoseconds*

*-Duration average per image: 1243747.2549999999 nanoseconds*

*Student image:*

*-Duration total: 295947495.0000000000 nanoseconds*

*-Duration average per image: 1479737.4750000001 nanoseconds*

*Testing constructor timing of image with constructing image of 5000x1 1000 times*

*Private image:*

*-Duration total: 6248216.0000000000 nanoseconds*

*-Duration average per image: 6248.2160000000 nanoseconds*

*Student image:*

*-Duration total: 6457019.0000000000 nanoseconds*

*-Duration average per image: 6457.0190000000 nanoseconds*

*Testing constructor timing of image with constructing image of 1x5000 1000 times*

*Private image:*

*-Duration total: 7465161.0000000000 nanoseconds*

*-Duration average per image: 7465.1610000000 nanoseconds*

*Student image:*

*-Duration total: 221940043.0000000000 nanoseconds*

*-Duration average per image: 221940.0430000000 nanoseconds*

Geef de meetresultaten overzichtelijk weer in de vorm van een tabel en/of diagram.

## Verwerking

Duidelijk zichtbaar is dat onze constructor slomer is dan de originele constructor.

Laat zien hoe je de meetresultaten verwerkt om een conclusie te kunnen trekken. Het is niet nodig om alle berekeningen op te schrijven, als je bijvoorbeeld maar laat zien welke formule(s) je gebruikt voor het verwerken van de meetresultaten en daar zo nodig één voorbeeldberekening aan toevoegt.

## Conclusie

Geef aan welke conclusie kan worden getrokken uit de verwerking van de meetresultaten.

## Evaluatie

Leg een verband tussen de getrokken conclusie en het doel van het experiment (en de hypothese). Ga daarbij ook in op bijvoorbeeld de meetonzekerheid als gevolg van de gebruikte meetmethoden of eventuele meetfouten.